

### **REMARKS**

Claims 1-30 are pending in the present application. By this Response, claims 1-30 are amended. Claims 1, 2, 9, 15, 16, 23, 29 and 30 are amended for clarity and to correct for antecedent basis. Claims 3-5, 10, 11, 17-19, 24 and 25 are amended to correct for antecedent basis. Claims 6-8, 12-14, 20-22 and 26-28 are amended in view of the amendments made to independent claims 1, 9, 15 and 23. Reconsideration of the claims is respectfully requested.

#### **I. 35 U.S.C. § 112, Second Paragraph**

The Final Office Action rejects claims 1-30 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Claims 1, 9, 15, 23, 29 and 30 are amended for clarity by deleting the unclear portions of the claims and splitting the "wherein" clause. Applicants respectfully request that these amendments be allowed into the record as the amendments only clarify previously claimed subject matter and do not introduce new subject matter. Claims 3-5, 10, 11, 17-19, 24 and 25 are amended to correct for antecedent basis. Claims 6-8, 12-14, 20-22 and 26-28 are amended in view of the amendments made to independent claims 1, 9, 15 and 23. Therefore, the rejection of claims 1-30 under 35 U.S.C. § 112, second paragraph is overcome.

#### **II. 35 U.S.C. § 103, Alleged Obviousness, Claims 1-30**

The Office Action rejects claims 1-30 under 35 U.S.C. § 103(a), as being allegedly unpatentable over Shackelford et al. (U.S. Patent No. 5,511,196) in view of Reuss et al. (U.S. Patent No. 5,579,318). This rejection is respectfully traversed.

As to claim 1, the Office Action states:

As per claim 1 Shackelford teaches a method of updating an object association between a source object and a target object (col. 3 line 61 –

col. 4 line 55). Shackelford does not specifically teach the use of updating a source to identify the target.

Reuss teaches the use of updating a target value holder of the target object to identify the source object, the target value holder comprising a value holder that the target object points to (col. 12 lines 25-35, col. 3 lines 46-56, col. 5 lines 39-42, col. 6 lines 27-45); and updating a source value holder of the source object to identify the target object, the source value holder comprising a value holder that the source object points to (col. 3 lines 46-56, col. 5 lines 39-42, col. 6 lines 27-45, col. 13 lines 5-24); and

wherein a value holder is an object that wrappers a target or source in order to implement a proxy pattern (col. 3 line 61 –col. 4 line 55).

It would have been obvious to one skilled in the art at the time of the invention to combine the teachings of Reuss and Shackelford because Reuss's updating of the source would improve Shackelford's object oriented system by being able to have not only the target but also the source with the most updated information.

Final Office Action dated June 4, 2004, page 4.

Claim 1, which is representative of the other rejected independent claims 15 and 29 with regard to similarly recited subject matter, reads as follows:

1. A method of updating an object association between a source object and a target object, comprising:
  - updating a target value holder of the target object with a new source to identify the source object, wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern; and
  - updating a source value holder of the source object with a new target to identify the target object, wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern. (emphasis added)

Shackelford and Reuss, taken alone or in combination, fail to teach or suggest updating a target value holder of the target object with a new source to identify the source object, wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern, and updating a source value holder of the source object with a new target to identify the target object, wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern.

Shackelford is directed to a system for the establishment of relationships between reference objects in an object oriented environment and an associated data object residing

outside an object oriented environment. In the Shackelford system, a data object within an application outside an object oriented environment is identified and then multiple reference objects within an oriented environment are established. Each reference object has a unique identifier that is associated with one of multiple users and is associated with the identified data object so that multiple users may concurrently access the data object utilizing an associated reference object. The associated data object may then be modified in response to a modification of any reference object and, similarly, the reference objects may be modified in response to a modification of the associated data object.

Thus, with the system of Shackelford, the association established between the reference object and the data object is not maintained within the object oriented environment as the data object is outside the object oriented environment. The present claims, taken in light of the specification, are directed to maintaining object associations in an object oriented environment. Thus, Shackelford is not directed to the same field of invention as the presently claimed invention. In addition, Shackelford does not teach or suggest updating a target value holder of the target object with a new source to identify the source object, wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern, and updating a source value holder of the source object with a new target to identify the target object, wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern. The Office Action dated December 18, 2003, alleges that these features are taught by Shackelford at column 3, line 61 to column 4, line 55, which reads as follows:

Multiple users, which may comprise actual users of other applications or systems, depicted by using object 48 and using object 50, within object oriented environment 52 may desire to access and modify data object 40, such as by deleting it or by sending it to another application. A master reference object 44 is established and associated with data object 40. Copies of the master reference object 44 are then established and associated with each user. For example, reference object 42 is created and associated with using object 48 while reference object 46 is created and associated with using object 50. Therefore, each user within an object oriented environment has an associated reference object within the object oriented environment which then is associated with a data object in an application outside the object oriented environment. It should

also be apparent that a reference object may be automatically created in response to a creation of a data object.

A reference object within an object oriented environment created in this manner may be associated with a data object external to the object oriented environment. Thereafter, when a user modifies a data object, all associated reference objects are also modified to reflect the new state of the data object. Similarly, when a user modifies a reference object, the associated data object is also modified to reflect the new state of the reference object.

Multiple object oriented environments and multiple applications may all exist within one data processing system. One object oriented environment may be linked to one or more applications at one time, and one application may be linked to one or more object oriented environments at one time.

Referring to FIG. 3, there is depicted a block diagram of multiple users and associated reference objects established in accordance with the method and system of the present invention. As described above, a copy of master reference object 44 is created and associated with each user. In this manner, using object 48 is associated with reference object 42, using object 50 is associated with reference object 44, and using object 60 is associated with reference object 62. Although the content of reference objects 42, 44, and 46 is the same, each object has a unique identifying name. Therefore, there is a one-to-one relationship between reference objects and using objects, so that it is easy to maintain control of the reference objects. In this manner, it is clear which reference object is being utilized each time a using object creates, modifies, or deletes a reference object.

With reference to FIG. 4, there is illustrated a block diagram depicting strong and weak reference relationships created in accordance with the method and system of the present invention. As illustrated, a strong reference relationship exists between reference object 42 and data object 40. When a strong relationship exists, there will be a validation within object oriented environment 52 that a reference object has an existing data object, and that a data object has an associated reference object.

The relationship between reference object 42 and data object 40, and between reference object 62 and data object 70 are strong relationships. Reference objects 42 and 62 will be maintained along with data objects 40 and 70. There will also be a validation that data objects 40 and 70 exist.

In this section, Shackelford is merely describing creating an association between reference objects within the object oriented environment to the data objects that are outside the object oriented environment. The Final Office Action admits that Shackelford does not teach updating a target value holder of the target object with a new

source to identify the source object and updating a source value holder of the source object with a new target to identify the target object; however, Applicants respectfully submit that Shackelford does not teach or suggest wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern and wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern.

The Office Action dated December 18, 2003 cites column 4, lines 50-54 as teaching this specific features, which reads as follows:

The relationship between reference object 42 and data object 40, and between reference object 62 and data object 70 are strong relationships. Reference object 42 and 62 will be maintained along with data objects 40 and 70. There will be a validation that the data objects 40 and 70 exist'.

This specific section of Shackelford merely teaches through creating the relationship of reference object 42 in the object oriented environment to data object 40 outside of the object oriented environment and through creating the relationship of reference object 62 in the object oriented environment to data object 70 outside of the object oriented environment and that the reference objects 42 and 62 will be maintained along with data objects 40 and 70. This section does not teach a value holder which is an object that wrappers an instance of another object in order to implement a proxy pattern. Shackelford does not teach or suggest implementing a proxy pattern nor does Shackelford teach or suggest wrapping an instance of an object. Thus, in addition to the Final Office Action admitting that Shackelford does not teach updating a target value holder of the target object with a new source to identify the source object and updating a source value holder of the source object with a new target to identify the target object, Shackelford does not teach or suggest wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern and wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern.

Additionally, the Final Office Action alleges that Reuss teaches updating a target value holder of the target object with a new source to identify the source object at column

12, lines 25-35, column 3, lines 46-56, column 5, lines 39-42 and column 6, lines 27-45, which read as follows:

FIG. 15 shows logic design flow for updating a Target Service Control Point from a Source. The Target receives a Sync Update Message. If the relevant local data is newer than that contained in the message, then a NewerData Result Message is built, the local data is included in the Result Message, and that message is returned to the Source. If the local data is older, then it is updated and time stamped; a Success Result Message is built and sent to the Source as well. If the effort to update the Target data fails, the Result Message is set to "failure" and that message is returned to the Source Service Control Point, a Resync Process is scheduled at the Source (as shown in FIGS. 11 and 12) and the process ended (as shown in FIG. 11). If a second attempt to send a Success Result Message is unsuccessful, the process is ended or a Rcsync Process is scheduled (as shown in FIG. 11).

(Column 12, lines 25-35)

One Network Element in a pair that contains redundant databases updates and thus reconciles with its mate Network Element according to the present invention. For instance, according to a preferred embodiment of the present invention, and at a database level, when a service application has received subscriber changeable data, its Service Control Point identifies that data as changed and causes corresponding information to be transmitted to the relevant application in the mate Service Control Point together with (preferably but not necessarily) information about the sending application address, receiving application address, subscriber identification and data identification.

(Column 3, lines 46-56)

If, for instance, a first Service Control Point or application ("Source") receives subscriber changeable data from a Service Node that is desired to be the same in both SCP's or applications, that data constitutes Synchronous Data with which the Source updates its mate application or Service Control Point ("Target").

(Column 5, lines 39-42)

In the preferred embodiment, the Service Node logic collects the data from the subscriber, verifies it and passes control to the update part of logic which performs the following steps: (a) constructs a routing string for the particular subscriber which is employed in the destination Service Control Point for routing; (b) sends an update Request Message to either one of the Service Control Points (preferably with an origination time stamp which the Service Node has generated for use with all updating efforts on the Network Elements for that particular data); (c) if the update

is a success, provides information to the subscriber and continues call processing; (d) if the update is a failure, attempts to send the update to the other (or another) Service Control Point; (e) if the second attempt is a success, provides information to the subscriber and continues call processing; and (f) if the second update fails, denies the service change to the subscriber and terminates the call. Other originating entities may just as easily employ similar and other steps for updating.

(Column 6, lines 27-45)

Reuss is directed to a data concurrence system for maintaining concurrence of parallel sets of data stored in telecommunication Network Elements. Though Reuss mentions the terms target and source, the Reuss system is a telecommunication system and not an object oriented environment. Furthermore, in column 12, lines 25-35, Reuss is merely teaching updating a Target Service Control Point (TSCP) from a Source. The TSCP is a network element within the telecommunications network and is updated when a message, which includes data to update the TSCP is sent from the source. If the relevant local data in the TSCP is newer than that contained in the message, then a NewerData Result Message is built, the local data is included in the Result Message, and that message is returned to the Source. If the local data is older, then it is updated and time stamped; a Success Result Message is built and sent to the Source as well. In column 3, lines 46-56, Reuss is merely teaching that network elements in pairs contain redundant database updates and reconcile between themselves. In column 5, lines 39-42, Reuss is merely teaching a telecommunications network that contains a source and a target. In column 6, lines 27-45, Reuss is merely teaching updating telecommunication network element information between a target network element and a source network element. None of these sections teach or suggest updating a target value holder of the target object with a new source to identify the source object.

Furthermore, Reuss does not teach updating a source value holder of the source object with a new target to identify the target object. The Final Office Action alleges that this feature is taught at column 3, lines 46-56, column 5, lines 39-42, column 6, lines 27-45, all shown above, and column 13, lines 5-24, which reads as follows:

1. A process for maintaining data concurrence in a plurality of databases in a telecommunications network, including the steps of:
  - a. providing to a Control Network Element information regarding data in at least one of the databases which is desired to be changed;

- b. communicating to a Source Network Element that contains at least one of the databases, via the Control Network Element, a first message that contains (i) information about the data that is desired to be changed; and (ii) a time stamp;
- c. changing information in the database in the Source Network Element based on information in the message received by the Source Network Element;
- d. communication to a Target Network Element that contains at least one of the databases, via the Source Network Element, a second message that contains (i) information about the data that is desired to be changed; and (ii) the time stamp; and
- e. changing information in the database in the Target Network Element based on information in the message received by the Target Network Element.

Column 3, lines 46-56, column 5, lines 39-42, column 6, lines 27-45 have been addressed above. Column 13, lines 5-24 merely summarizes the teaching of Reuss which are updating elements between network elements in a telecommunications network not in an object oriented environment. Thus, Reuss does not teach or suggest updating a target value holder of the target object with a new source to identify the source object and updating a source value holder of the source object with a new target to identify the target object. Moreover, Reuss does not teach or suggest wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern and wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern.

The Final Office Action alleges that Reuss teaches these features at column 3, line 61 to column 4, line 55, which reads as follows:

If successful, the first Service Control Point sends additional data (or in the event of an error, retransmits the earlier data). If the returned data is erroneous for a predetermined number of attempts, an error report is generated and synchronization efforts are terminated and may be rescheduled for later. If there is an error response or no reply from the mate Service Control Point for a predetermined number of attempts, the Service Control Point assumes that its mate (or mate application) is out of service and then schedules a resynchronization process to occur a predetermined time later. If the resynchronization process runs a predetermined number of times without success, an error report is generated and the audit is terminated or rescheduled.

As discussed below, the processes and systems of the present invention preferably occur at platform level rather than at application level



in order, among other things, to realize economies of scale, avoid multiple copies of code and the need to reconcile them, make development easier, and avoid degradation of call processing functionality by eliminating unnecessary competition for processor capacity when service applications need it the most.

Additionally, a control network element according to the present invention may include or add time related data (hereinafter sometimes referred to as a "time stamp") in or to a requested data change, so that this single stamp may be employed by other network elements involved in synchronization and other processes according to the present invention. This approach avoids the conventional need for network elements to be precisely time-synchronized, even when at remote locations, in order for proper data reconciliation to happen.

It is, furthermore, preferable that processes of the present invention occur across a non-overlay network, such as (but not limited to) SS7.

It is accordingly an object of the present invention to provide improved processes and systems for maintaining data concurrence in redundant information infrastructure databases, which processes and systems operate quickly, efficiently, reliably and may be implemented in existing platform based software modules and across existing communication links so as to impose minimal additional processing or communications loads.

It is an additional object of the present invention to eliminate confusion that otherwise occurs when one network element simultaneously attempts to update two others and is forced to reconcile situations when updates have failed on either or both network elements.

It is an additional object of the present invention to eliminate the necessity of ensuring that network elements are time-synchronized with each other in order for their databases properly to be maintained concurrent.

It is a further object of the present invention to provide data concurrence processes which increase reliability of call processing by, among other things, reducing opportunities for Service Control Points or applications to miss receipt of subscriber changeable data.

It is a further object of the present invention to provide resynchronization processes which cause data to be synchronized before the point in time at which they are subject to query by the network.

Nowhere in this section, or any other section of Reuss, is a target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern and a source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern taught or suggested. Thus, Reuss does not teach or suggest updating a target value holder of the target object with a new source to identify the source object, wherein the target value holder is an object that wrappers an

instance of the source object in order to implement a proxy pattern and updating a source value holder of the source object with a new target to identify the target object, wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern.

Moreover, neither reference teaches or suggests the desirability of incorporating the subject matter of the other reference. That is, there is no motivation offered in either reference for the alleged combination. The Final Office Action alleges that the motivation for the combination is "Reuss's updating of the source would improve Shackelford's object oriented system by being able to have not only the target but also the source with the most updated information." However, neither reference teaches or suggests a target value holder that is an object that wrappers an instance of the source object in order to implement a proxy pattern and a source value holder that is an object that wrappers an instance of the target object in order to implement a proxy pattern. Thus, the only teaching or suggestion to even attempt the alleged combination is based on a prior knowledge of Applicants' claimed invention thereby constituting impermissible hindsight reconstruction using Applicants' own disclosure as a guide.

One of ordinary skill in the art, being presented only with Shackelford and Reuss, and without having a prior knowledge of Applicants' claimed invention, would not have found it obvious to combine and modify Shackelford and Reuss to arrive at Applicants' claimed invention. To the contrary, even if one were somehow motivated to combine Shackelford and Reuss, and it were somehow possible to combine the two systems, the result would not be the invention, as recited in claim 1.

Even assuming, arguendo, that one would be motivated to combine Shackelford and Reuss, the combination would not result in the presently claimed invention. Instead, the combination would result in a system that creates relationships between a reference object in an object oriented environment and a data object outside a object oriented environment, as in Shackelford, and a system that updates network element in a telecommunications network, as in Reuss. However, Shackelford and Reuss, taken individually or in combination, do not teach updating a target value holder of the target object with a new source to identify the source object, wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy

pattern, and updating a source value holder of the source object with a new target to identify the target object, wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern.

Independent claims 9, 23 and 30 recite similar subject matter to independent claims 1, 15 and 29. Claim 9, which is representative of the other rejected independent claims 23 and 30 with regard to similarly recited subject matter, recites "updating a source value holder of the source object to identify the target object wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern, and updating a target value holder of the target object to identify the source object, wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern."

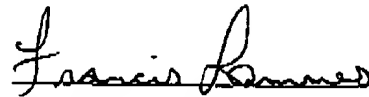
Thus, Shackelford and Reuss, taken alone or in combination, fail to teach or suggest all of the features in independent claims 1, 9, 15, 23, 29 and 30. At least by virtue of their dependency on claims 1, 9, 15 and 23, Shackelford and Reuss, taken alone or in combination, fail to teach or suggest all of the features of dependent claims 2-8, 10-14, 16-22 and 24-28. Accordingly, Applicants respectively request withdrawal of the rejection of claims 1-30 under 35 U.S.C. § 103(a).

**III. Conclusion**

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: August 4, 2004



Francis Lammes  
Reg. No. 55,353  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 367-2001  
Agent for Applicants